

**KONVEYOR CERDAS DENGAN FITUR PEMILAH
BERDASAR WARNA, PENIMBANG BERAT, DAN
PEMANTAUAN JUMLAH BARANG BERBASIS IOT**



**Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada
Jurusan Teknik Elektro Fakultas Teknik**

Oleh:

YULVI HIDAYATI

D400181153

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2020**

HALAMAN PERSETUJUAN

KONVEYOR CERDAS DENGAN FITUR PEMILAH BERDASAR WARNA, PENIMBANG BERAT, DAN PEMANTAUAN JUMLAH BARANG BERBASIS IOT

PUBLIKASI ILMIAH

oleh:

YULVI HIDAYATI
D400181153

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing



Ir. Pratomo-Budi Santosa, M.T
NIK : 627

HALAMAN PENGESAHAN

KONVEYOR CERDAS DENGAN FITUR PEMILAH BERDASAR WARNA, PENIMBANG BERAT, DAN PEMANTAUAN JUMLAH BARANG BERBASIS IOT

OLEH

YULVI HIDAYATI

D400181153

Telah dipertahankan di depan Dewan Penguji
Fakultas Teknik Jurusan Teknik Elektro
Universitas Muhammadiyah Surakarta
Pada hari Selasa, 14 Januari 2020
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. Ir. Pratomo Budi Santosa, M.T
(Ketua Dewan Penguji)
2. Bambang Hari Purwoto, S.T., M.T
(Anggota I Dewan Penguji)
3. Mochammad Muslich, ST, M.Eng
(Anggota II Dewan Penguji)



Dekan,

Ir. Sri Sunarjono, M.T.Ph.D.

NIK. 628

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam naskah publikasi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 14 Januari 2020

Penulis



YULVI HIDAYATI

D400181153

KONVEYOR CERDAS DENGAN FITUR PEMILAH BERDASAR WARNA, PENIMBANG BERAT, DAN PEMANTAUAN JUMLAH BARANG BERBASIS IOT

Abstrak

Konveyor Cerdas Dengan Fitur Sortir Berdasar Warna, Penimbang Berat, Serta Pemantauan Jumlah Barang Berbasis IoT ini dibuat dengan beberapa tahapan yaitu studi literatur, perancangan sistem, perancangan mekanis, perancangan elektronis, perancangan perangkat lunak, pembuatan alat, dan pengujian alat. Sistem pemilahan menggunakan sensor warna TCS3200 dengan proses pemilahan menggunakan servo. Warna yang akan dipilah terdiri dari dua warna yaitu merah dan hijau. Berat barang ditimbang dengan proses penimbangan menggunakan sensor *loadcell* dan modul hx711. Data jumlah barang hasil pemilahan dapat dipantau dari komputer ataupun *smartphone* yang sudah terinstal *web browser*. Modul wifi ESP8266 digunakan untuk menghubungkan alat ke jaringan internet. Perangkat-lunak yang digunakan dalam pembuatan IoT adalah XAMPP (Apache *web server* dan MySQL). Semua komponen tersebut dikontrol dan diprogram menggunakan mikrokontroler Arduino Uno. Berdasarkan hasil pengujian, sistem konveyor ini dapat memilah dua warna, yaitu merah dan hijau dengan tingkat keberhasilan pembacaan sensor warna untuk warna merah dan hijau sebesar 90%. Konveyor ini dapat melakukan penimbangan berat objek barang dengan persentase galat sebesar 1,31%. ESP8266 membutuhkan waktu untuk terhubung ke *server*, berdasarkan hasil pengujian respon waktu, waktu tunggu terlalu lama sekitar 35 detik dan tercepat 1 detik.

Kata Kunci: *Internet of Things*, konveyor, *loadcell*, pemilah, TCS3200.

Abstract

Intelligent Conveyor With Color-Based Sorting, Weight Weighing, And IoT-Based Goods Quantity Monitoring is made in several stages, namely literature study, system design, mechanical design, electronic design, software design, tool manufacturing, and tool testing. The sorting system uses the TCS3200 color sensor with the servo sorting process. The color to be sorted consists of two colors namely red and green. Item weight is weighed by the weighing process using a *loadcell* sensor and hx711 module. The data of the number of goods separated can be monitored from a computer or smartphone that has a web browser installed. The ESP8266 wifi module is used to connect devices to the internet network. The software used in making IoT is XAMPP (Apache web server and MySQL). All these components are controlled and programmed using an Arduino Uno microcontroller. Based on the test results, this conveyor system can sort out two colors, namely red and green with a success rate of color sensor readings for red and green by 90%.. This conveyor can weigh object goods with an error percentage of 1,31%. ESP8266 requires time to connect to the server, based on the results of testing the response time, the longest waiting time is around 35 seconds and the fastest is 1 second.

Keywords: Internet of Things, conveyors, *loadcell*, sorters, TCS3200.

1. PENDAHULUAN

Seiring dengan kemajuan teknologi saat ini yang dapat memudahkan serta mempercepat pekerjaan, manusia dituntut agar selalu kreatif dalam mengembangkan teknologi disegala aspek kehidupan tak terkecuali dalam dunia industri. Otomasi dalam industri dilakukan untuk mengurangi kesalahan manusia (*human error*), karena manusia cenderung menjadi lelah dan tidak efektif ketika melakukan pekerjaan terus-menerus untuk waktu yang lama (**Huang, 2010**). Saat ini, industri berusaha meminimalkan pekerjaan manusia, misalnya dalam tugas memisahkan. Pekerjaan memisahkan ini dilakukan terus-menerus, membutuhkan tingkat akurasi yang tinggi, dan sulit dilakukan oleh manusia selama 24 jam. Oleh karena itu, dibutuhkan media untuk mempermudah pekerjaan tersebut (**Ridha, 2016**).

Saat ini, dunia tengah memasuki era revolusi industri 4.0 atau revolusi industri dunia ke-empat. Revolusi industri 4.0 sudah memasuki dunia digitalisasi sistem industri, yang menekankan pada *digital*, *artificial intelligence*, dan *robotic*. Revolusi ini mendorong sistem otomatisasi dalam suatu proses, dan menekankan integrasi antar alat menggunakan internet. Data pada proses produksi diinputkan dalam sistem digital, sehingga dapat diakses di mana pun untuk mempermudah pencatatan hasil produksi.

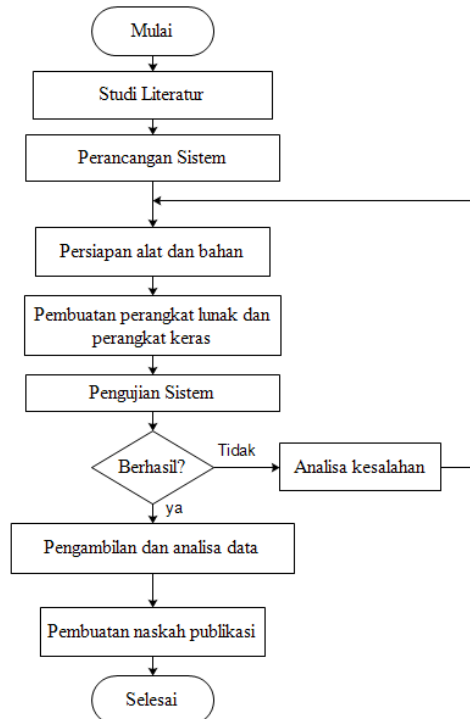
Salah satunya topik yang saat ini sedang dikembangkan dan banyak sekali yang bisa ditemukan aplikasinya di dunia industri adalah konveyor. Konveyor adalah suatu alat yang sering dijumpai pada industri, yang berfungsi memindahkan barang dari suatu tempat ke tempat yang lain yang jumlahnya banyak dan berkelanjutan, sehingga mampu memobilisasi barang dan lebih bernilai ekonomi dibandingkan menggunakan transportasi berat seperti truk dan mobil pengangkut. Konveyor telah mengalami perubahan tidak hanya memobilisasi barang, tetapi juga dapat difungsikan untuk hal lain dengan penambahan fitur seperti pemantauan, penimbangan, penghitung dan lain sebagainya.

Berdasarkan latar belakang tersebut, optimasi kinerja konveyor sangat dibutuhkan dalam penunjang kegiatan industri maju dengan menambahkan fitur-

fitur tambahan. Oleh karena itu pada penelitian ini dibuatlah suatu alat purwarupa konveyor yang dapat menimbang barang serta memilahnya berdasarkan warna dan memantau jumlah hasil produksi menggunakan teknologi IoT, sehingga dapat diakses melalui *web browser* dengan koneksi internet.

2. METODE

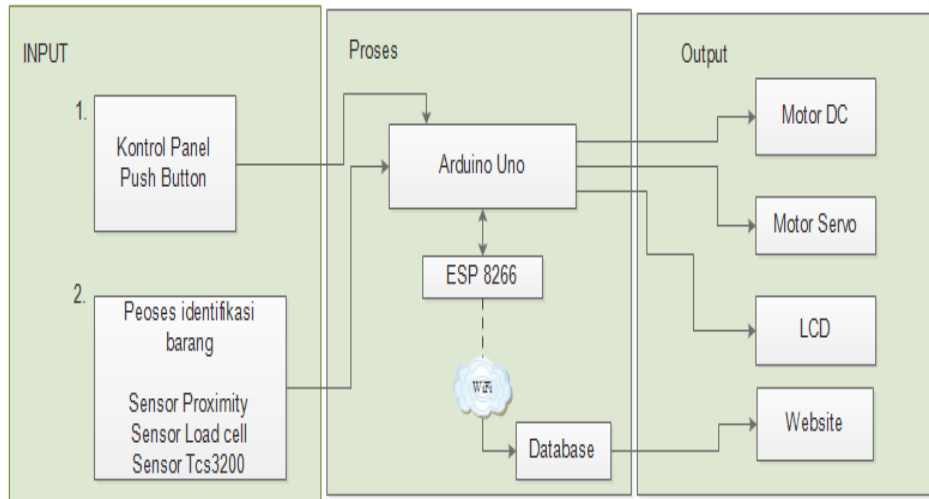
Untuk mendapatkan hasil penelitian yang baik, diperlukan langkah–langkah penelitian yang tepat dan sistematis. Pada pelaksanaan tugas akhir ini, secara garis besar langkah kerjanya terdiri atas tahap studi literatur, yaitu menghimpun data-data atau informasi-informasi yang berkaitan dengan topik tugas akhir. Informasi tersebut biasanya diperoleh dari buku–buku, jurnal–jurnal, serta artikel di internet yang terpercaya dan isinya dapat dipertanggung jawabkan. Selanjutnya dilakukan perancangan sistem, persiapan alat dan bahan, perancangan perangkat keras dan lunak, pembuatan dan implementasi alat, tahap pengujian alat, pengambilan dan analisis data, dan terakhir pembuatan laporan akhir naskah publikasi. Secara garis besar, urutan yang akan dilaksanakan dalam penelitian pada diperlihatkan pada Gambar 1.



Gambar 1. Alur penelitian

2.1 Diagram Blok Sistem

Secara keseluruhan kerja sistem konveyor dengan fitur pemilah berdasar warna, penimbang berat, dan pemantauan jumlah barang berbasis IoT diperlihatkan pada Gambar 2.



Gambar 2. Diagram Blok Sistem

Dari diagram blok sistem, terlihat ada tiga blok, yaitu *input*, proses, dan *output*. Arduino Uno sebagai mikrokontroler menjadi pusat sistem yang mengatur semua interaksi yang terjadi, pada blok *input* terdapat kontrol panel yang memberikan perintah untuk menjalankan sistem, proses identifikasi barang dari segi warna dan berat dilakukan oleh sensor yang menjadi *inputan* Arduino Uno untuk selanjutnya diproses dan menghasilkan perintah atau *output* berupa gerakan motor DC untuk menggerakkan sabuk konveyor, motor servo untuk memilah barang, serta LCD (Liquid Crystal Display) untuk menampilkan warna dan berat barang. Modul ESP8266 sebagai jembatan komunikasi yang mengkoneksikan Arduino Uno ke jaringan WiFi, sehingga dapat menerima dan mengirim data ke *server*. Data-data pembacaan sensor akan dikirimkan dan disimpan pada *database MySQL*, untuk ditampilkan pada halaman *web* dengan program PHP (*Hypertext Preprocessor*).

2.2 Perancangan Perangkat Keras

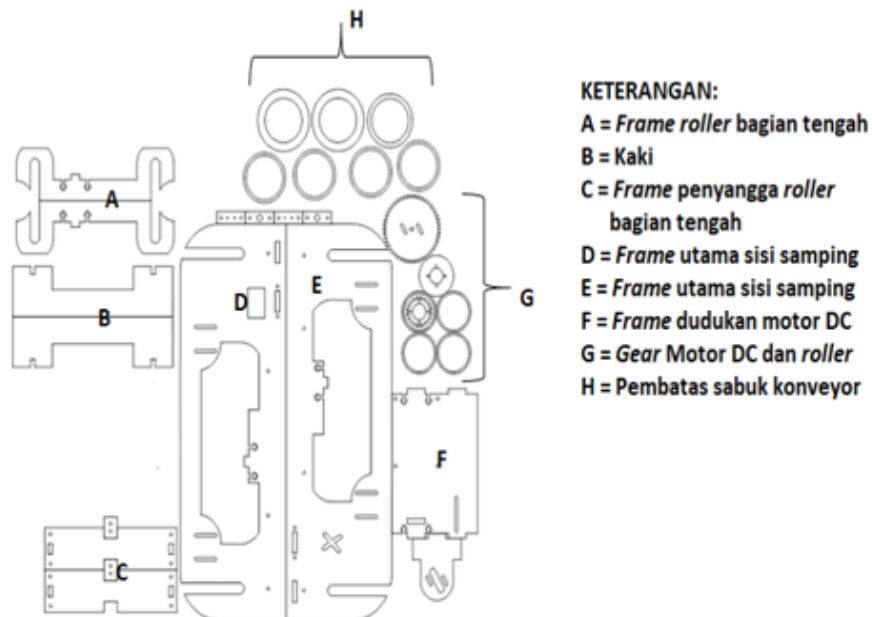
Perancangan perangkat keras ini terdiri atas dua bagian, yaitu perancangan elektronis dan perancangan mekanis.

2.2.1 Perancangan Elektronis

Perancangan desain elektronis dimaksudkan guna menggabungkan komponen elektronika yang digunakan sehingga terlihat lebih sederhana dan efisien. Perangkat-lunak yang digunakan dalam pembuatan perancangan ini adalah EAGLE. Untuk mendesain perancangan elektronis ini yang perlu diperhatikan adalah *wiring diagram* antar komponen terutama pada Arduino sebagai otak yang mengendalikan seluruh sistem kerja pada alat ini. Perancangan elektronis dapat dilihat pada lampiran 1 skematik rangkaian keseluruhan, dan lampiran 2 *board* rangkaian.

2.2.2 Perancangan Mekanis

Perancangan Mekanis dimaksudkan guna mendesain kerangka perangkat keras, sehingga dapat menyerupai keadaan sebenarnya. Dalam hal ini menggunakan *software* CorelDraw untuk mendesain kerangka konveyor dengan bahan kerangka berupa akrilik 5mm. Sedangkan bagian tambahan lainnya seperti kotak panel, penggeser servo, tabung penyimpanan barang, dan kotak barang yang akan menjadi objek pemilahan dan penimbangan, didesain menggunakan aplikasi *solidwork* dan dicetak menggunakan 3D printer dengan bahan filament. Desain kerangka konveyor dapat dilihat pada Gambar 3.



Gambar 3. Desain Kerangka Konveyor

Roller merupakan penumpu utama dari barang yang ditransportasikan, menggunakan bahan berbentuk tabung yaitu pipa paralon yang juga difungsikan sebagai penyanggah sabuk berbahan karet. Kerangka *roller* dibagian tengah berfungsi sebagai penyanggah *roller* di tengah *frame* utama, sehingga posisinya tidak berpindah-pindah. Kaki atau tiang penyangga difungsikan sebagai pondasi untuk badan konveyor. *Frame* utama merupakan rangka badan yang menopang semua *roller* termasuk bagian tengah agar *roller* tidak berpindah-pindah. *Frame* dudukan motor DC digunakan untuk meletakkan motor DC. Motor DC dipasang *gear*, dan *roller* yang ada di samping motor DC juga dipasangkan *gear*, sehingga ketika motor DC bergerak *roller* juga ikut bergerak dan menggerakkan sabuk konveyor.

2.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri dari atas bagian, yaitu perancangan *web* dan perancangan program mikrokontroler.

2.3.3 Perancangan *web*

Perancangan ini bertujuan untuk menentukan isi konten yang akan ditampilkan pada halaman *web*. Pada tugas akhir ini, isi konten yang akan ditampilkan meliputi halaman *login*, halaman utama berupa hasil pemantauan data barang, dan halaman *logout*. Halaman pemantauan barang berisi data barang yang masuk berupa hasil *request* data berat, warna, waktu dan *id*. Bahasa pemograman yang digunakan dalam pembuatan *web* ini adalah PHP, HTML, dan lain-lain. Tahap perancangan *web* terdiri atas database dan tampilan *web*.

1. Database

Pembuatan database dilakukan di phpmyadmin dengan membuat 2 tabel yaitu tabel *user* untuk *login*, dan tabel barang untuk pemantauan.

a. Tabel user

Merupakan tabel untuk menyimpan data pengguna, tabel user terdiri dari 4 atribut, yaitu *userID*, *userName*, *userEmail*, dan *userPassword* yang ditunjukkan pada Tabel 1.

Tabel 1. Rancangan Tabel *User*

Atribut	Tipe Data	Keterangan
<i>UserID</i>	Int(11)	Id pengguna
<i>UserName</i>	Varchar(50)	Nama pengguna
<i>UserEmail</i>	Varchar(50)	Email pengguna
<i>UserPassword</i>	Varchar(50)	Password pengguna

b. Tabel barang

Merupakan tabel untuk menyimpan data barang yang berasal dari pembacaan sensor yang akan dipantau, dan ditampilkan pada halaman *web*, Tabel barang terdiri atas 4 atribut, yaitu *id*, waktu, berat, dan warna yang ditunjukkan pada Tabel 2.

Tabel 2. Rancangan Tabel Barang

Atribut	Tipe Data	Keterangan
Id	Int(10)	Id barang
Waktu	timestamp	Waktu masuk barang
Berat	double	Berat barang
Warna	enum('Merah', 'Hijau')	Warna barang

2. Tampilan *web*

Tampilan *web* memanfaatkan program *open source* yang di-*download* dari *github* untuk mendapatkan file css sebagai *template* website. Teks editor yang digunakan untuk perancangan *web* adalah *sublime text*.

2.3.4 Perancangan Program Mikrokontroler

Mikrokontroler yang digunakan adalah Arduino Uno, sehingga perlu diprogram dan di compile menggunakan teks editor Arduino IDE. Program ini merupakan bagian yang paling penting untuk menjalankan sistem sesuai yang diinginkan, sehingga penulisan algoritmanya harus benar. Program arduino yang dibuat di tugas akhir ini dibagi beberapa sub bagian untuk memudahkan penulisan dan jika terjadi eror. Gambar perancangan program mikrokontroler dapat dilihat pada Gambar 4.

```
konveyor_iot | Arduino 1.8.1
File Edit Sketch Tools Help

konveyor_iot SoftwareSerial mode_kalibrasi mode_run mode_settings tombol_dan_menu

#include <Wire.h>
#include<LiquidCrystal_I2C.h>
#include<Servo.h>
#include<SoftwareSerial.h>
LiquidCrystal_I2C lcd(0x27 , 16, 2);
#include "HX711.h"
HX711 scale(A3, A2);
#define S0 6
#define S1 7
#define S2 12
#define S3 13
#define color_out 5
#define In1 2
#define In2 4
#define En 3
#define tomboll_atas 600
#define tomboll_bawah 400
#define tombol2_atas 100
#define tombol2_bawah 30
// #define prox_threshold 750
#define interval_button 1000
#define tolerance_val 30
// #define loadcell_calibration 2230
Servo ambil, pilih;
String ip_address = "192.168.43.176", ssid = "yulvi", password = "yulvihi97";
float units_gram, final_gram, gram_correct, sum_gram;
```

Gambar 4. Perancangan Program mikrokontroler

3. HASIL DAN PEMBAHASAN

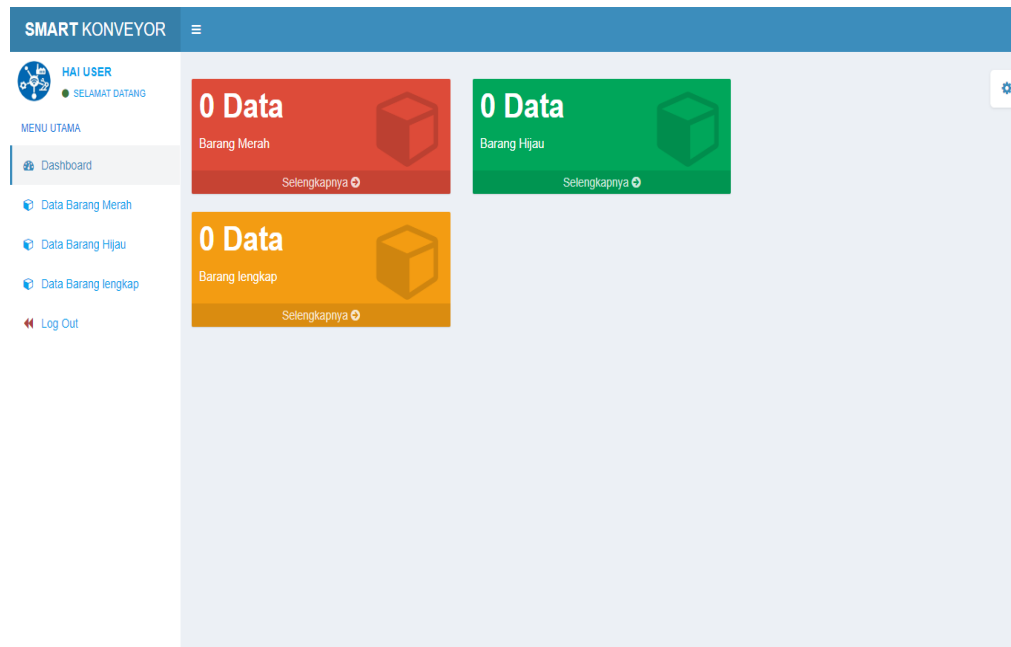
Bab ini memaparkan hasil rancangan yang telah dibuat sebelumnya disertai pengujian integrasi antara perangkat-keras dan perangkat-lunak. Hasil pengujian akan di analisis untuk mengidentifikasi kelebihan dan kelemahan sistem.

3.1 Hasil Rancangan dan Sistem Kerja Secara Keseluruhan

Setelah dilakukan perancangan perangkat-keras dan perangkat-lunak, bentuk fisik konveyor cerdas pemilah berdasar warna , penimbang berat dan Pemantauan jumlah barang berbasis IoT dapat dilihat pada Gambar 5, dan Gambar 6 memperlihatkan tampilan *web*.



Gambar 5. Bentuk Fisik Konveyor



Gambar 6. Tampilan *Web*

Mekanisme kerja alat ini ketika sistem dinyalakan atau dihubungkan ke sumber ESP8266 akan terhubung ke jaringan wifi, setelah berhasil terhubung LCD akan menampilkan menu. Tekan tombol “ok” selama minimal dua detik untuk menampilkan beberapa pilihan menu tersebut. Menu tersebut terdiri atas: 1. *Run*, 2. *Setting*, 3. Kalibrasi

Pilihan *run* untuk menjalankan sistem konveyor. Ketika tombol *run* ditekan dan ditahan beberapa detik, maka sabuk konveyor akan digerakkan oleh motor DC (*direct current*) dengan PWM (*pulse width modulation*) sebesar 200. Servo pengambil yang berada di bawah penampungan barang akan mengambil barang dan menggesernya ke sabuk konveyor. Ketika barang sudah terdeteksi oleh sensor *proximity*, maka konveyor akan berhenti. Sensor warna akan mengambil data warna barang dan sensor *loadcell* akan mengambil data berat barang. Data tersebut akan menjadi satu data *request* yang dikirim ke *server*.

Selanjutnya, ESP8266 akan melakukan koneksi ke akses point *server* lokal dengan alamat IP (*internet protocol*) yang sudah dideklarasikan pada program Arduino. Setelah *port* terbuka ESP8266 melakukan *get request* pada *server*

dengan mengirimkan data yang telah diambil oleh sensor. Setelah menerima respon “Ok” data sudah berhasil terkirim pada *server*, servo pemilah akan bergerak sesuai warna yang terdeteksi pada sensor warna. Konveyor akan mengarahkan barang ke servo pemilah, sehingga barang berhasil dipilah berdasar warna. Sistem ini akan berjalan secara terus-menerus, untuk mengembalikan sistem ke menu dapat dilakukan dengan menekan tombol *back* atau reset.

Pilihan *setting* difungsikan untuk mengatur beberapa parameter, dengan menekan tombol pada kotak panel tanpa harus membuka program Arduino. Pengaturan tersebut antara lain seperti jumlah sampel data pembacaan berat barang. Hal ini dibuat karena pembacaan *loadcell* yang tidak stabil, sehingga dengan pengaturan ini pembacaan berat diambil sebanyak jumlah yang diinginkan, kemudian diambil nilai rata-ratanya, sehingga dengan ini pembacaan menjadi lebih akurat. Parameter yang diatur selanjutnya adalah kecepatan motor DC, yaitu dengan menambahkan nilai PWM driver motor L298N dengan kecepatan maksimal 255. Selanjutnya, pengaturan pembacaan *proximity*. *Proximity* tersebut terbuat dari sensor LDR yang ditembakkan laser di depannya, pembacaan ADC (*Analog to Digital Converter*) sensor LDR sebelum ada objek barang dan setelah ada objek barang yang menghalangi pancaran laser di depan sensor LDR yang menjadi acuan deteksi barang. Nilai ADC sensor LDR akan semakin tinggi ketika intensitas cahaya kecil, dan akan turun ketika intensitas cahaya tinggi. Pengaturan dilakukan dengan memasukkan nilai acuan maksimum pembacaan sensor (*threshold*) ketika tidak ada barang yang terdeteksi atau ketika tidak ada barang yang menghalangi laser, pengaturan ini sangat dibutuhkan karena intensitas cahaya di tiap ruangan yang berbeda-beda. Pilihan kalibrasi bertujuan untuk mengkalibrasi pembacaan *loadcell* dan kalibrasi dilakukan saat konveyor dalam keadaan diam.

3.2 Pengujian Sensor Warna

Pada sensor warna TCS3200 terdapat selektor S2 dan S3 yang berfungsi untuk memilih kelompok konfigurasi fotodiode Konfigurasi pin S2 dan S3 dapat dilihat pada Tabel 3.

Tabel 3. Konfigurasi Pin S2 dan S3

S2	S3	Tipe Fotodioda
L	L	<i>Red</i>
L	H	<i>Blue</i>
H	L	<i>Clear</i>
H	H	<i>Green</i>

Fotodioda akan mengeluarkan arus yang besarnya sebanding dengan warna dasar cahaya yang menyimpannya. Arus tersebut dikonversikan menjadi sinyal kotak atau pulsa digital yang frekuensinya sebanding dengan nilai arusnya. Pada tugas akhir ini program arduino hanya mengaktifkan konfigurasi fotodioda merah dan hijau saja. Keluaran frekuensi tersebut diatur skalanya pada pin S1 dan S0. Penskalaan frekuensi dapat dilihat pada Tabel 4.

Tabel 4. Konfigurasi pin S1 dan S0

S0	S1	Skala Frekuensi
L	L	<i>Power down</i>
L	H	2 %
H	L	20 %
H	H	100 %

Untuk pengujian ini diatur penskalaan frekuensi sebesar 20%, dengan mengatur *pulseIn* program arduino pada kondisi *Low*, sehingga sensor akan membaca berapa banyak gelombang *Low* dalam suatu periode waktu. Jarak antara objek barang dengan sensor warna ± 1 cm. Hasil pengujian sensor warna dapat dilihat pada Tabel 5.

Tabel 5. Pengujian Sensor Warna

NO	Warna Barang	Berhasil Terdeteksi	Pembacaan nilai R(Red) ,G(Green) B(Blue)		
			R	G	B
1	Merah	Ya	52	118	86
2	Merah	Ya	54	117	86
3	Merah	Ya	72	117	92
4	Merah	Ya	62	123	92
5	Merah	Ya	59	123	83
6	Hijau	Ya	121	54	63
7	Hijau	Ya	121	53	64
8	Hijau	Ya	130	61	63
9	Hijau	Ya	147	71	83
10	Hijau	Tidak	143	57	54

Berdasarkan Tabel 5, warna diperoleh dari nilai keluaran skala frekuensi RGB (*Red, Green, Blue*). Pada program diatur *pulseIn Low*, maka nilai RGB terkecil diidentifikasi sebagai hasil pembacaan warna, misalnya warna merah diperoleh dengan membandingkan pembacaan RGB didapatkan nilai yang paling kecil adalah nilai R (*Red*), sedangkan warna hijau nilai yang paling kecil adalah G (*Green*).

Berdasarkan dari sepuluh data pengujian warna merah dan warna hijau yang masing-masing berjumlah lima data tiap warna, sensor berhasil mendeteksi sesuai dengan yang diinginkan sebanyak sembilan data percobaan, dengan satu kali gagal dibagian warna hijau.

Persentase keberhasilan pengujian sensor warna dapat dirumuskan menurut persamaan berikut ini.

$$\frac{\text{Jumlah data benar}}{\text{jumlah data pengujian}} \times 100\% \quad (1)$$

Sehingga berdasarkan hasil data pengujian sensor warna tingkat keberhasilan pembacaan sensor adalah 90%.

3.3 Pengujian Servo Bagian Pemilah

Pengujian servo bagian pemilah dimaksudkan untuk mengetahui tingkat keberhasilan gerakan servo ketika ada objek barang. Untuk objek barang warna merah, pulsa servo diatur sebesar 2000 μs dan objek barang warna hijau sebesar 1000 μs . Hasil sepuluh data pengujian dapat dilihat pada Tabel 6.

Tabel 6. Pengujian Servo Bagian Pemilah

NO	Warna Barang	Pulsa Servo (microsecond)	Berhasil
1	Merah	2000	Ya
2	Merah	2000	Ya
3	Merah	2000	Ya
4	Merah	2000	Ya
5	Merah	2000	Ya
6	Hijau	1000	Ya
7	Hijau	1000	Ya
8	Hijau	1000	Ya
9	Hijau	1000	Ya
10	Hijau	-	Tidak

Berdasarkan Tabel 6, servo berhasil bergerak sesuai pulsa yang diberikan dan sesuai dengan warna objek barang yang diidentifikasi oleh sensor warna. Dari sepuluh data tersebut, terdapat sembilan data berhasil, dan satu kali gagal. Gerakan servo tersebut sangat bergantung pada pembacaan sensor warna, sehingga tingkat keberhasilan pengujian ini adalah 90%.

3.4 Kalibrasi *loadcell*

Kalibrasi dilakukan dengan cara membandingkan nilai pembacaan sensor *Loadcell* dengan alat ukur, kalibrasi ini dilakukan saat konveyor dalam kondisi diam untuk mendapatkan hasil yang lebih akurat. Dari kalibrasi ini nanti diketahui nilai galat dari pembacaan *loadcel* pada konveyor, nilai galat merupakan selisih antara berat asli dengan nilai pembacaan sensor *loadcell*, proses kalibrasi dengan mengambil beberapa variasi data berat, dari variasi data tersebut nilai galat akan di persentase dengan rumus persentase galat sebagai berikut ini.

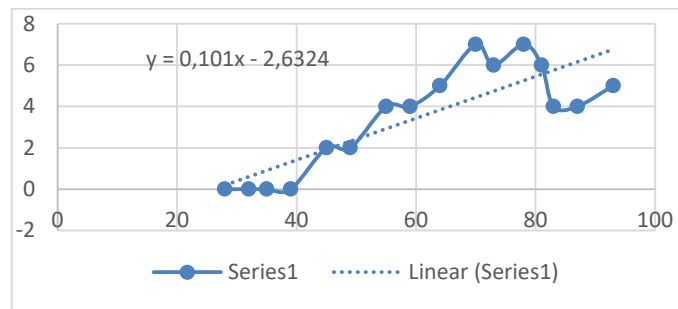
$$\frac{[berat\ asli - pembacaan\ loadcell]}{berat\ asli} * 100\% \quad (2)$$

Hasil kalibrasi *loadcell* dengan mengambil beberapa variasi data berat dapat dilihat pada Tabel 7

Tabel 7. Data Kalibrasi *Loadcell*

BERAT ASLI (GRAM)	PENGUKURAN LOADCELL (GRAM)	GALAT	PERSENTASE GALAT
28	28	0	0
32	32	0	0
35	35	0	0
39	39	0	0
43	45	2	4,65
47	49	2	4,25
51	55	4	7,84
55	59	4	7,27
59	64	5	8,47
63	70	7	11,11
67	73	6	8,95
71	78	7	9,85
75	81	6	8
79	83	4	5,06
83	87	4	4,81
88	93	5	5,68
RATA-RATA PESENTASE GALAT =			5,37 %

Nilai rata-rata persentase galat yang didapatkan sebesar 5,37 %. Data pada Tabel 7 dapat dicari kembali persamaan faktor koreksi untuk mendapatkan nilai yang lebih akurat, dengan cara membuat grafik perbandingan nilai galat dengan pembacaan *loadcell*. Sumbu X diisi dengan nilai pembacaan *loadcell* dan sumbu Y diisi nilai galat, diperoleh persamaan linear $y = 0,101x - 2,6324$ persamaan ini digunakan pada program Arduino sebagai faktor koreksi pembacaan *loadcell*. Hasil perbandingan dapat dilihat pada Gambar 7.



Gambar 7. Grafik Perbandingan Nilai Galat dan Pembacaan Loadcell

3.5 Pengujian *Loadcell* Setelah Kalibrasi

Pengujian *loadcell* setelah kalibrasi adalah pengujian setelah memasukkan persamaan yang diperoleh dari kalibrasi sebelumnya, sehingga diperoleh data dari nilai pembacaan sensor *loadcell* seperti pada Tabel 8.

Tabel 8. Pengujian *Loadcell* Setelah Kalibrasi

BERAT ASLI (GRAM)	PEMBACAAN LOADCELL	GALAT	PERSENTASE GALAT
28	28	0	0
32	31	1	3,13
35	34	1	2,86
39	38	1	2,56
43	43	0	0
47	47	0	0
51	52	1	1,96
55	56	1	1,81
59	60	1	1,69
63	64	1	1,58
67	67	0	0
71	70	1	1,41
75	77	2	2,66
79	80	1	1,27
83	83	0	0
88	88	0	0
RATA-RATA PESENTASE GALAT =			1,31 %

Berdasarkan Tabel 8, diperoleh rata-rata persentase galat sebesar 1,31%. Tahap pengujian ini dilakukan saat sabuk konveyor posisi diam atau mode kalibrasi.

3.6 Pengujian Respon waktu

Pengujian ini dilakukan untuk mengetahui seberapa lama waktu barang untuk bisa sampai diujung proses pemilahan. Data pengujian respon waktu dapat dilihat pada Tabel 9.

Tabel 9. Pengujian Respon Waktu

No	Warna Barang	Waktu mulai pengidentifikasian warna dan pengiriman data ke <i>server</i>	Waktu sampai barang setelah dipilah	Waktu tunggu
1.	Hijau	13:59:53	14:00:28	35 detik
2.	Hijau	14:02:37	14:03:11	34 detik
3.	Hijau	14:03:31	14:03:47	16 detik
4.	Merah	14:05:55	14:06:15	20 detik
5.	Merah	14:06:41	14:07:02	21 detik
6.	Merah	14:07:22	14:07:23	1 detik

Berdasarkan Tabel 9, waktu tunggu barang untuk mengidentifikasi warna dan mengirim data ke *server* terlama adalah 35 detik, sedangkan respon tercepat adalah 1 detik. Hal ini terjadi karena ESP8266 butuh waktu untuk mengkoneksikan ke *server*. Ketika hubungan terputus maka ESP8266 akan memerlukan waktu untuk menghubungkan lagi ke *server*. Akan tetapi, jika proses hubungan belum *timeout* dan ada barang, maka ESP8266 akan langsung mengirimkan data.

3.7 Pengujian Barang dengan Warna Lain

Warna yang dipilah dari sistem konveyor ini ada dua, yaitu merah dan hijau, sehingga perlu adanya pengujian warna lain untuk mengetahui sistem konveyor sudah berjalan sesuai program atau belum. Servo pemilah diatur pulsa sebesar 1500 μ s ketika barang tidak terdeteksi atau di luar warna yang telah ditentukan.. Hasil pengujian barang untuk warna lain, dapat dilihat pada Tabel 10.

Tabel 10. Pengujian Barang Dengan Warna Lain

No	Warna Barang	Pulsa Servo (microsecond)	Berhasil dideteksi	Pembacaan RGB		
				R	G	B
1	Hitam	1500(diam posisi tengah)	Tidak	209	236	179
2	Putih	1500(diam posisi tengah)	Tidak	15	16	12
3	Merah muda	2000	Ya	21	34	24
4	Kuning	2000	Ya	28	45	52
5	Biru	1500(diam posisi tengah)	Tidak	83	51	35







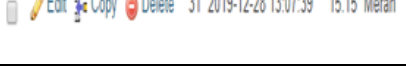

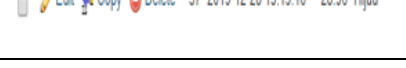
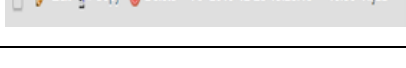
Berdasarkan dari hasil pengujian dengan warna lain seperti hitam, putih, merah muda, kuning dan biru, diperoleh nilai perbandingan RGB. Warna merah muda dan kuning nilai RGB terkecil adalah nilai *Red*, sehingga warna tersebut masih terdeteksi sebagai barang warna merah. Hal tersebut terjadi karena sensor warna diprogram hanya mengaktifkan konfigurasi pin fotodiode *Red* dan *Green* saja, dengan membandingkan nilai RGB terkecil sebagai identifikasi warna, keadaan ini sangat berpengaruh ketika diuji dengan warna selain warna merah dan hijau.

Pada program arduino diatur keadan barang yang terdeteksi adalah warna hijau dan merah sehingga jika ada warna lain dengan nilai RGB terkecil nilai *Blue*, maka barang tidak akan terdeteksi seperti yang terlihat dari tabel 10. Warna hitam, putih dan biru nilai RGB terkecilnya adalah *Blue* sehingga warna-warna tersebut tidak terdeteksi, dan keadaan servo sesuai dengan yang diinginkan yaitu diam pada posisi tengah, saat warna tersebut tidak terdeteksi.

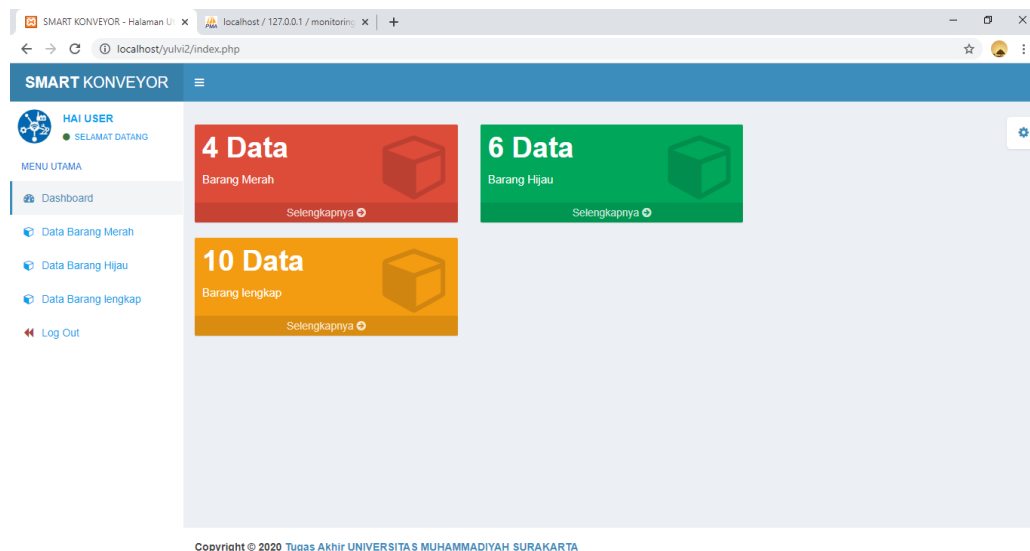
3.8 Pengujian Pengiriman data

Untuk mengetahui data *request* pembacaan sensor berat dan warna berhasil dikirim ke database *server* dilakukan pengujian pengiriman data. Proses pengiriman data menggunakan ESP8266 dengan perintah *AT command*. Data hasil pengujian dapat dilihat pada Tabel 11.

Tabel 11. Pengujian Pengiriman Data

No	Poses pengiriman data	Hasil Pada Database
1	12:58:18.451 -> 1. at command => AT+CIPSEND=124 OYI 12:58:18.484 -> 2. at command => AT+CIPCLOSE=0 Fail 12:58:24.666 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 12:58:24.700 -> GET /yulvi2/add_script.php?id=46&berat=35.23&input_warna='Hijau'	 4 2019-12-28 12:58:18 35.23 Hijau
2.	12:58:48.662 -> 0. at command => AT+CIPSEND=124 OYI 12:58:48.695 -> 1. at command => AT+CIPCLOSE=0 OYI 12:58:49.791 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 12:58:49.825 -> GET /yulvi2/add_script.php?id=56&berat=18.45&input_warna='Hijau'	 5 2019-12-28 12:58:48 18.45 Hijau
3.	13:00:07.807 -> 1. at command => AT+CIPSEND=124 OYI 13:00:07.873 -> 2. at command => AT+CIPCLOSE=0 OYI 13:00:08.968 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 13:00:09.002 -> GET /yulvi2/add_script.php?id=86&berat=24.99&input_warna='Merah'	 8 2019-12-28 13:00:07 24.99 Merah
4.	13:01:38.387 -> 0. at command => AT+CIPSEND=125 OYI 13:01:38.453 -> 1. at command => AT+CIPCLOSE=0 Fail 13:01:44.633 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 13:01:44.666 -> GET /yulvi2/add_script.php?id=136&berat=29.01&input_warna='Merah'	 13 2019-12-28 13:01:38 29.01 Merah
5.	13:04:39.540 -> 1. at command => AT+CIPSEND=125 OYI 13:04:39.573 -> 2. at command => AT+CIPCLOSE=0 OYI 13:04:40.670 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 13:04:40.703 -> GET /yulvi2/add_script.php?id=22&berat=17.69&input_warna='Hijau'	 22 2019-12-28 13:04:39 17.69 Hijau
6.	13:04:56.814 -> 4. at command => AT+CIPSEND=125 OYI 13:04:56.847 -> 5. at command => AT+CIPCLOSE=0 OYI 13:04:57.911 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 13:04:57.977 -> GET /yulvi2/add_script.php?id=236&berat=31.67&input_warna='Merah'	 23 2019-12-28 13:04:56 31.67 Merah
7.	13:07:35.654 -> 0. at command => AT+CIPSEND=125 OYI 13:07:35.687 -> 1. at command => AT+CIPCLOSE=0 Fail 13:07:42.267 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 13:07:42.333 -> GET /yulvi2/add_script.php?id=316&berat=15.15&input_warna='Merah'	 31 2019-12-28 13:07:39 15.15 Merah
8.	13:14:54.285 -> 2. at command => AT+CIPSEND=125 OYI 13:14:54.318 -> 3. at command => AT+CIPCLOSE=0 OYI 13:14:55.414 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 13:14:55.480 -> GET /yulvi2/add_script.php?id=47&berat=14.64&input_warna='Hijau'	 47 2019-12-28 13:14:54 14.64 Hijau
9.	13:19:16.884 -> 3. at command => AT+CIPSEND=125 OYI 13:19:16.917 -> 4. at command => AT+CIPCLOSE=0 OYI 13:19:18.047 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 13:19:18.080 -> GET /yulvi2/add_script.php?id=57&berat=28.96&input_warna='Hijau'	 57 2019-12-28 13:19:16 28.96 Hijau
10.	13:25:18.238 -> 0. at command => AT+CIPSEND=125 OYI 13:25:18.271 -> 1. at command => AT+CIPCLOSE=0 OYI 13:25:19.334 -> selesaiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii 13:25:19.401 -> GET /yulvi2/add_script.php?id=76&berat=16.36&input_warna='Hijau'	 76 2019-12-28 13:25:18 16.36 Hijau

Berdasarkan dari hasil pengujian pengiriman data, sistem telah berhasil mengirimkan data pada database dengan keterangan waktu, berat,dan warna yang sama. Pengiriman tersebut menggunakan *request GET* pada skrip PHP. Ketika berhasil diterima, maka data *request* tersebut di masukkan ke database. Hasil dari penyimpanan data di database akan ditampilkan pada halaman *website* yang dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Web

4. PENUTUP

4.1 Kesimpulan

Setelah melakukan perancangan, perakitan dan pengujian pada Tugas Akhir ini, dapat diambil beberapa kesimpulan sebagai berikut:

1. Sensor warna telah berhasil mendeteksi 2 warna yaitu warna merah dan hijau sesuai objek barang, dengan dengan jarak pembacaan dari barang ± 1 cm, tingkat keberhasilan sebesar 90% untuk kedua warna tersebut.
2. Pembacaan sensor berat saat kondisi konveyor diam dengan persentase rata-rata galat sebesar 1,31 %.
3. ESP8266 telah berhasil mengirim data *request* ke database, menggunakan perintah *AT command*.
4. Hasil pengujian pada warna selain warna merah dan hijau, sensor membaca nilai RGB dan membandingkannya, ketika nilai RGB yang terkecil adalah R atau G walaupun objek barang tersebut bukan warna merah atau hijau, maka barang tetap akan terdeteksi. Sedangkan jika nilai RGB terkecilnya B, maka barang tidak akan terdeteksi.
5. ESP8266 membutuhkan waktu tunggu untuk terhubung ke *server*, dengan waktu tunggu terlama sekitar 35 detik dan tercepat 1 detik.

4.2 Saran

1. Sensor warna diprogram hanya mengaktifkan konfigurasi pin fotodioda *Red* dan *Green* saja, dengan membandingkan nilai RGB terkecil sebagai identifikasi warna, hal ini sangat berpengaruh ketika diuji dengan warna selain warna merah dan hijau, sehingga untuk warna lain perlu pengambilan dari banyak sampel warna sehingga diketahui range RGB tiap-tiap warna.
2. Mekanis sangat mempengaruhi pembacaan *loadcell*, saat konveyor berjalan pembacaan *loadcell* tidak bisa akurat dan stabil, sehingga perlu adanya perbaikan pada penelitian selanjutnya.
3. Sistem IoT konveyor ini hanya pada *server* komputer lokal dan untuk mengaksesnya harus satu jaringan internet dengan WiFi yang terhubung oleh ESP8266 dengan mengakses IP komputer *server* lokal tersebut. Diharapkan untuk penelitian selanjutnya dapat dibuat *public* dengan menyewa *hosting*, sehingga dapat diakses di mana saja walaupun tidak satu jaringan internet dengan ESP8266.
4. *Request* ESP8266-01 ke server terlalu lama, padahal suatu industri membutuhkan kinerja yang cepat, sehingga untuk penelitian selanjutnya bisa menggunakan modul wifi lainnya atau dengan *firmware* versi terbaru untuk kinerja yang cepat.

PERSANTUNAN

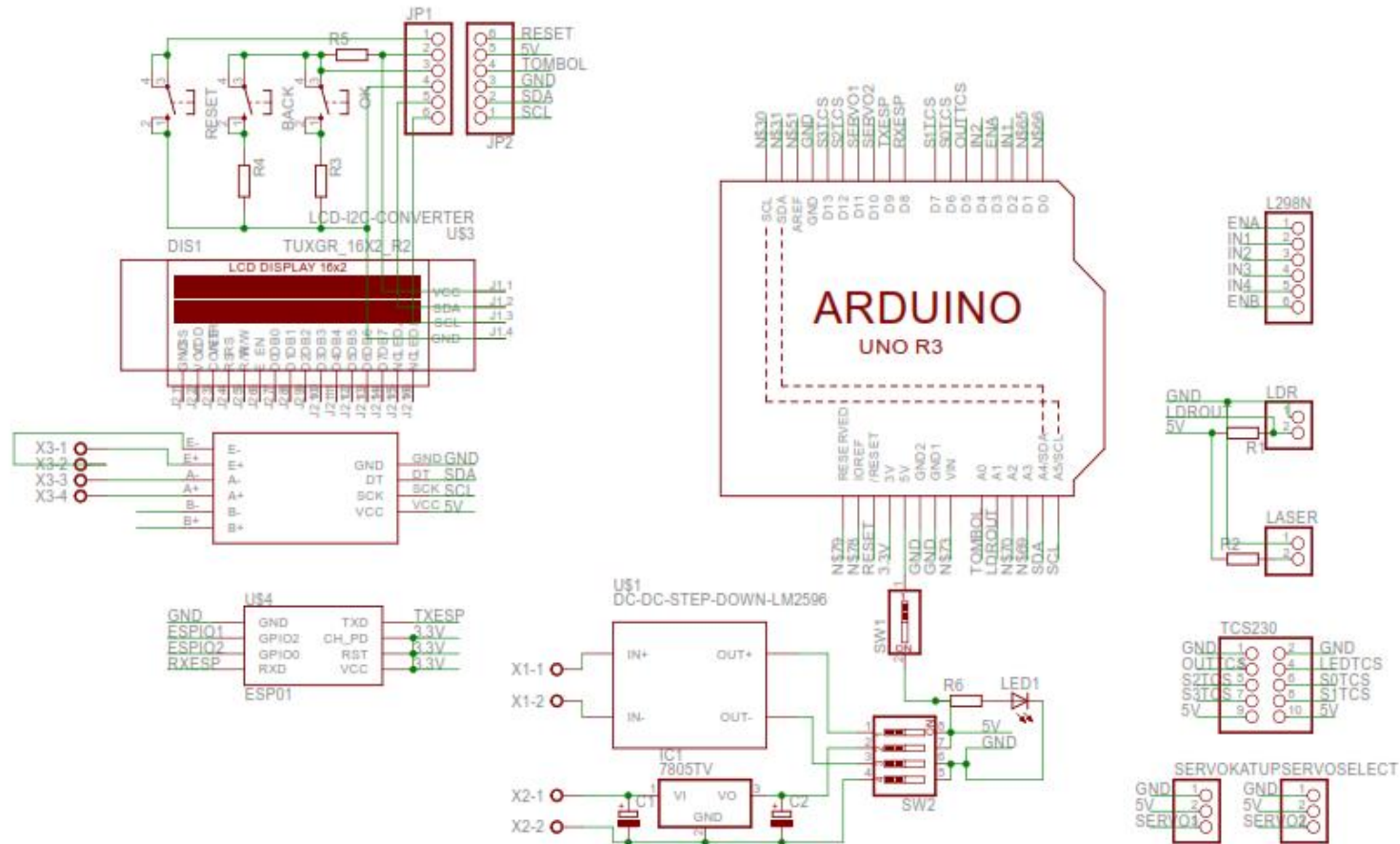
Alhamdulillah segala puji bagi Allah Subhanahu wata'ala yang telah memberikan nikmat karunia-nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.

DAFTAR PUSTAKA

- Anhar, 2010. PHP & MySql Secara Otodidak. Jakarta: PT TransMedia
Huang, Z. 2011. Analysis of Hu's Moment Invariants on Image Scaling and Rotation. ECU Publ., no. 2010, pp. 476–480, 2011.
Loves, Erik Firmanto Da (2017) Prototipe Pemilah Benda Berdasarkan

- Bentuk dan Warna menggunakan *conveyor*. Teknik Elektro Universitas Sanata Dharma.
- Mahali, M. 2016, Smart Door Locks Based On Internet of Things Concept With Mobile Backend as a Service. *Jurnal Electronic, Informatics, and Vocational Education (ELINVO)*, 1(3), 171-181.
- Marantika, D. (2017). Timbangan Digital Berbasis Nodemcu ESP8266 I2E Via Broker IOT Tweer, Tugas Akhir, D3 Elektronika dan Instrumentasi, Sekolah Vokasi, Universitas Gajah Mada.
- Ridha, M. 2016. Prototype Konveyor Seleksi Objek Berdasarkan Warna Menggunakan Kamera Pixy CMUCAM 5 Berbasis Arduino. Teknik Elektro Universitas Muhammadiyah Surakarta.
- Sholihin, Achmad., 2010, MySQL 5 dari Pemula Hingga Mahir, Jakarta : Universitas Nudi Luhur
- Utdirartatmo, 2002, Mengelola Database Server MySQL di Linux dan Windows, Yogyakarta : Andi Publisher
- Zhou, Q., & Zhang, J. (2011). Internet of things and geography review and prospect. *Proceedings - 2011 International Conference on Multimedia and Signal Processing, CMSP 2011*, 2, 47–51.

LAMPIRAN 1 SKEMATIK RANGKAIAN



LAMPIRAN 2 BOARD RANGKAIAN

